CS10720 Problems and Solutions

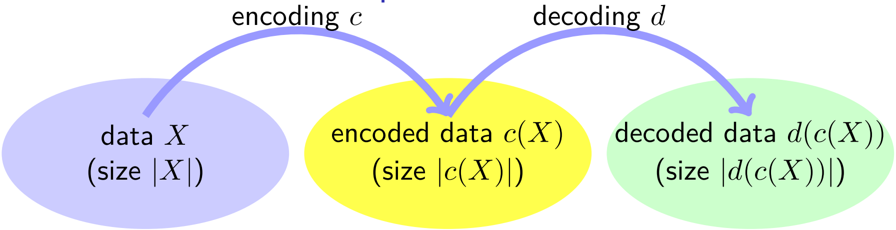
Thomas Jansen

Today: Lossless Compression
Revision

April 25th

Plans for Today

Remember: Lossless Compression



Lossless Compression $\quad d(c(X)) = X$ for all texts X

Letter-wise compression

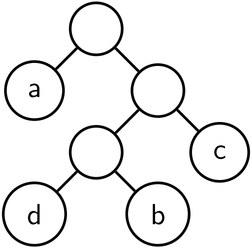
$X = x_1 x_2 \cdots x_l \colon c(X) = c(x_1)c(x_2)\cdots c(x_l)$

Information theory (remember Claude Shannon)

- Assumption text $X \in \Sigma^*$ comprises of letters $s \in \Sigma$ with each s occuring with fixed, independent probability $\mathrm{Prob}\,(s)$
- Definition entropy $-\sum_{s\in\Sigma} \mathrm{Prob}\,(s) \log \mathrm{Prob}\,(s)$
- Result average coding length bounded below by entropy

Huffman Coding

Huffman Coding example



for

s	$c(s)$
a	0
b	101
c	11
d	100

Facts Huffman coding has optimal expected length and expected length $\leq$ entropy $+ 1$

Huffmann Coding and Entropy

Remember Huffman coding has
optimal expected length and
expected length $\leq$ entropy $+ 1$

How much can '$+1$' hurt?

Example $\Sigma = \{\mathsf{a}, \mathsf{b}\}, \mathrm{Prob}\,(\mathsf{a}) = .99, \mathrm{Prob}\,(\mathsf{b}) = .01$
Huffman code $c(\mathsf{a}) = 0$, $c(\mathsf{b}) = 1$
expected length $1 \cdot .99 + 1 \cdot .01 = 1$
entropy $-.99 \log(.99) - .01 \log(.01) \leq .081$
absolute redundancy expected length $-$ entropy $\geq .919$
Observation relative redundancy $\geq 1234\%$

Idea improve by considering sequences of length k as letters
(increasing alphabet size to $|\Sigma|^k$)

Fact expected length $\leq$ entropy $+ 1/k$

Alternative: Arithmetic Coding

Idea encoding $c \colon \Sigma^* \to [0,1)$ injective

Consider alphabet $\Sigma = \{s_1, s_2, \ldots, s_n\}$ with $\mathrm{Prob}(s_i)$ (given)

Define $F(j) = \sum_{i=1}^{j} \mathrm{Prob}(s_i)$

Encoding Partition $[0,1)$ into intervals $[0, F(1))$, $[F(1), F(2))$, …, $[F(n-1), F(n))$.
Map first letter s_i into $[F(i-1), F(i))$.
Partition interval $[F(i-1), F(i))$ proportionally like above.
Map second letter into these sub-intervals
and continue like this.

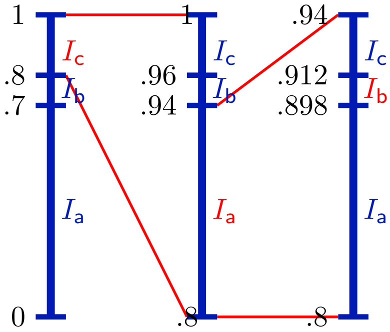
Decoding of code $c \in [0,1)$

1. $l_0 := 0$; $u_0 := 1$; $i := 0$
2. Repeat until decoding complete:
3. Determine $s_k \in \Sigma$ with $c \in [F(k-1), F(k))$.
4. $l_{i+1} := F(k-1)$; $u_{i+1} := F(k)$
5. $c := (c - l_{i+1})/(u_{i+1} - l_{i+1})$; $i := i + 1$

Example Arithmetic Encoding

Given $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$, $\mathrm{Prob}(\mathsf{a}) = .7$, $\mathrm{Prob}(\mathsf{b}) = .1$, $\mathrm{Prob}(\mathsf{c}) = .2$
text cab

Intervals $I_{\mathsf{a}} = [0, .7)$, $I_{\mathsf{b}} = [.7, .8)$, $I_{\mathsf{c}} = [.8, 1)$



Observation

any $v \in [.898, .912)$ encodes cab
e. g., $c(\mathsf{cab}) = .905$

Example Arithmetic Decoding

Given $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$, $\mathrm{Prob}(\mathsf{a}) = .7$, $\mathrm{Prob}(\mathsf{b}) = .1$, $\mathrm{Prob}(\mathsf{c}) = .2$
code $.905$

Intervals $I_{\mathsf{a}} = [0, .7)$, $I_{\mathsf{b}} = [.7, .8)$, $I_{\mathsf{c}} = [.8, 1)$

Decoding of code $c \in [0, 1)$

1. $l_0 := 0$; $u_0 := 1$; $i := 0$
2. Repeat until decoding complete:
3. Determine $s_k \in \Sigma$ with $c \in [F(k-1), F(k))$.
4. $l_{i+1} := F(k-1)$; $u_{i+1} := F(k)$
5. $c := (c - l_{i+1})/(u_{i+1} - l_{i+1})$; $i := i + 1$

c	$.905$	$.525$	$.75$
i	0	1	2
l	0	$.8$	0
u	1	1	$.7$
text	c	a	b

On Arithmetic Coding

Remember arithmetic coding maps Σ^* injectively onto $[0,1)$
with efficient coding and decoding
almost without additional overhead (like coding table)

Facts when treating sequences of length k as single letters

Huffman coding	expected length $\leq$ entropy $+ 1/k$
arithmetic coding	expected length $\leq$ entropy $+ 2/k$

Remember Huffman coding requires additional Huffman tree with $|\Sigma|^k$
$\rightsquigarrow$ arithmetic coding more practical

Example actual application lossless compression for bitmap images
JBIG (http://www.jpeg.org/jbig/)

Other Compression Methods

- dictionary approaches
  - idea store frequent words in dictionary; encode by their index
  - implementation using either fixed or adaptive dictionaries; using 'sliding window' approaches for good dictionaries
  - applications zip, compress, GIF, ...
- list update
  - idea start with arbitrary ordering of letters; encode letter by position; (potentially) change ordering after each letter by moving it to front
  - implementation use clever transform (Burrows-Wheeler Transform (BWT)) to group equal letters; compute RLE; use arithmetic encoding for RLE
  - application bzip2
- lossy compression
  - idea do not store irrelevant information
  - observation 'irrelevant' highly depending on context; for images 'hard to see', for audio 'hard to hear'
  - applications jpeg, mp3, MPEG, ...

Exam Revision

Remember CS107 assessment comprises of three elements

1. portfolio 40%
2. in-class test 30%
3. exam 30%

You have **either** already passed
by accumulating $\geq 40\%$ in the portfolio and in-class test
and want to improve your mark
or have not yet passed
and need the exam to pass (and perhaps a bit more)

In both cases you're interested in doing well in the exam

(If you're not interested in doing well in the exam
you're really just wasting your time being here.)

About the Exam

The exam

- contributes 30% to the overall mark
- is marked on a scale of 0–30
- has a duration of 2 hours
- will ask you to answer THREE out of FOUR questions
- will not allow the use of calculators
  (because you won't need them)
- will allow you to bring any printed material
  Opinion slides or lecture notes probably most useful

Please

- before the exam, prepare for the exam
- in the exam, attempt to answer at least three questions
  (even if you know little, you may get marks for an attempt)

Today example questions and potential answers
demonstrating potential exam questions
(not real future questions, obviously)

Integers: Comparing Integers

Problem For each of the four representations (sign value; one's complement; two's complement; excess with bias 7), sort the four bit strings in ascending order when read as integers in those representation.
0110; 1111; 1000; 0001

Remember sign-value representation (2016-01-28:15–18)
leftmost bit is sign, other bits standard binary encoding

Towards a solution

- positive numbers $>$ negative numbers
- $0110 > 0001$
- $111 > 000$, therefore $1111 < 1000$

Solution $1111 < 1000 < 0001 < 0110$

Integers: Comparing Integers

Problem For each of the four representations (sign value; one's complement; two's complement; excess with bias 7), sort the four bit strings in ascending order when read as integers in those representation.
0110; 1111; 1000; 0001

Remember one's complement representation (2016-02-01:24–28)
leftmost bit signals sign
leftmost bit 0 ⇒ other bits standard binary encoding
leftmost bit 1 ⇒ complement of other bits standard binary

Towards a solution

- positive numbers > negative numbers
- $0110 > 0001$
- $111 > 000$, therefore $1111 > 1000$

Solution $1000 < 1111 < 0001 < 0110$

Integers: Comparing Integers

Problem For each of the four representations (sign value; one's complement; two's complement; excess with bias 7), sort the four bit strings in ascending order when read as integers in those representation.
0110; 1111; 1000; 0001

Remember two's complement representation (2016-02-01:29–33)
leftmost bit signals sign
leftmost bit 0 $\Rightarrow$ other bits standard binary encoding
leftmost bit 1 $\Rightarrow$ 'standard binary e. $- 2^{l-1}$'

Towards a solution

- positive numbers $>$ negative numbers
- $0110 > 0001$
- $111 > 000$, therefore $1111 > 1000$

Solution $1000 < 1111 < 0001 < 0110$

Integers: Comparing Integers

Problem For each of the four representations (sign value; one's complement; two's complement; excess with bias 7), sort the four bit strings in ascending order when read as integers in those representation.
0110; 1111; 1000; 0001

Remember excess representation (2016-02-01:34–36)
'standard binary e. − bias'

Towards a solution

- order exactly the same as in standard binary encoding

Solution $0001 < 0110 < 1000 < 1111$

Integers: Comparing Properties of Representations

Problem Compare two's complement and excess representation with respect to the following properties:

1. number of different numbers that can be represented
2. smallest and largest integer that can be represented
3. support of addition
4. support of comparisons

	two's complement	excess representation	remark
1	2^l	2^l	equal
2	$-2^{l-1}+1,\ldots,2^{l-1}$	$-b,\ldots,2^l-1-b$	can be equal
3	excellent for all numbers	very problematic	advantage two's c.
4	differences depending on sign	excellent for all numbers	advantage excess

Rationals: Comparing Numbers in IEEE754 Format

Problem Compare the following rational numbers (all represented in IEEE 754 format, binary with 32 bits). For each of the following three pairs of numbers decide if $<$, $=$ or $>$ is the correct comparison.

1. $1\,0000\,0000\,000\,0000\,0000\,0000\,0000\,0000$ and $0\,0000\,0000\,000\,0000\,0000\,0000\,0000\,0000$
2. $0\,1001\,1010\,101\,0001\,1000\,0000\,0000\,0000$ and $0\,1001\,1001\,111\,1111\,1111\,1111\,1111\,1111$
3. $0\,1000\,0001\,001\,0101\,1010\,0000\,0000\,0000$ and $1\,0011\,0110\,010\,1100\,0000\,0000\,0000\,0000$

Remember IEEE 754 representation (2016-02-04:55–61)

1. first number represents -0, second number represents $+0$
   Solution $=$
2. both numbers positive, first number with bigger exponent
   Solution $>$
3. first number positive, second number negative
   Solution $>$

IEEE 754 and Integer Representation

Problem Consider the different parts of the IEEE 754 representation and discuss which representations for integers you can find in the different components.

- components of IEEE 754 representation: exponent, sign, mantissa
- exponent uses excess representation
- mantissa is always non-negative number, sign is indicated by sign bit
  identical to sign value representation

Matrix Arithmetic

Problem Consider two 3×3 matrices $A = (a_{i,j})$ and $B = (b_{i,j})$ with $a_{i,j} = 2i + j$ and $b_{i,j} = 3j - i$.

1. Compute $A + B$.
2. Compute $-2 \cdot A$.
3. Compute $A \cdot B$.

Remember matrix addition (2016-02-25:160–161)
scalar matrix multiplication (2016-02-25:165–166)
matrix multiplication (2016-02-25:167–169)

1. $A = \begin{pmatrix} 3 & 4 & 5 \\ 5 & 6 & 7 \\ 7 & 8 & 9 \end{pmatrix}$, $B = \begin{pmatrix} 2 & 5 & 8 \\ 1 & 4 & 7 \\ 0 & 3 & 6 \end{pmatrix}$,

$$A + B = \begin{pmatrix} 3+2 & 4+5 & 5+8 \\ 5+1 & 6+4 & 7+7 \\ 7+0 & 8+3 & 9+6 \end{pmatrix}$$

Matrix Arithmetic

Problem Consider two 3×3 matrices $A = (a_{i,j})$ and $B = (b_{i,j})$ with $a_{i,j} = 2i + j$ and $b_{i,j} = 3j - i$.

1. Compute $A + B$.
2. Compute $-2 \cdot A$.
3. Compute $A \cdot B$.

Remember matrix addition (2016-02-25:160–161)
scalar matrix multiplication (2016-02-25:165–166)
matrix multiplication (2016-02-25:167–169)

2. $A = \begin{pmatrix} 3 & 4 & 5 \\ 5 & 6 & 7 \\ 7 & 8 & 9 \end{pmatrix}, -2 \cdot A = \begin{pmatrix} -2 \cdot 3 & -2 \cdot 4 & -2 \cdot 5 \\ -2 \cdot 5 & -2 \cdot 6 & -2 \cdot 7 \\ -2 \cdot 7 & -2 \cdot 8 & -2 \cdot 9 \end{pmatrix}$

Matrix Arithmetic

Problem Consider two 3×3 matrices $A = (a_{i,j})$ and $B = (b_{i,j})$ with $a_{i,j} = 2i + j$ and $b_{i,j} = 3j - i$.

1. Compute $A + B$.
2. Compute $-2 \cdot A$.
3. Compute $A \cdot B$.

Remember matrix addition (2016-02-25:160–161)
scalar matrix multiplication (2016-02-25:165–166)
matrix multiplication (2016-02-25:167–169)

3. $A = \begin{pmatrix} 3 & 4 & 5 \\ 5 & 6 & 7 \\ 7 & 8 & 9 \end{pmatrix}$, $B = \begin{pmatrix} 2 & 5 & 8 \\ 1 & 4 & 7 \\ 0 & 3 & 6 \end{pmatrix}$, $A \cdot B =$

$$\begin{pmatrix} 3\cdot 2 + 4\cdot 1 + 5\cdot 0 & 3\cdot 5 + 4\cdot 4 + 5\cdot 3 & 3\cdot 8 + 4\cdot 7 + 5\cdot 6 \\ 5\cdot 2 + 6\cdot 1 + 7\cdot 0 & 5\cdot 5 + 6\cdot 4 + 7\cdot 3 & 5\cdot 8 + 6\cdot 7 + 7\cdot 6 \\ 7\cdot 2 + 8\cdot 1 + 9\cdot 0 & 7\cdot 5 + 8\cdot 4 + 9\cdot 3 & 7\cdot 8 + 8\cdot 7 + 9\cdot 6 \end{pmatrix}$$

Summary & Take Home Message ‘Compression’

Things to remember

- arithmetic coding
- other compression methods

Take Home Message

- Compression is a fascinating topic with practical applications in many areas.
- Lossless compression is well understood. There may still be room for improvements in practice.
- Lossy compression is less settled. But introducing new standards is hard.

Summary & Take Home Message ‘Exam Preparation’

Things to remember

- exam contributes 30% $\Rightarrow$ no panic
- numbers
- matrices

Take Home Message

- Don’t panic.
- Please, prepare for the exam.
- Not all topics are relevant for the exam. What is relevant is covered in the example problems on Blackboard and the revision lectures.